

2005

Intentional Embodied Agents

Alan Martin
University College Dublin

Gregory M.P. O'Hare
University College Dublin

Brian R. Duffy
University College Dublin

John F. Bradley
University College Dublin

Bianca Schoen-Phelan
Technological University Dublin, bianca.phelan@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/scschcomcon>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Martin, A., O'Hare, G., Duffy, B., Bradley, J. & Schon-Phelan, B. (2005). Intentional Embodied Agents. *18th International Conference on Computer Animation and Social Agents (CASA 2005)*, Hong Kong, China, October 17-25, 2005.

This Conference Paper is brought to you for free and open access by the School of Computing at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact yvonne.desmond@tudublin.ie, arrow.admin@tudublin.ie, brian.widdis@tudublin.ie.



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 License](#)

Intentional Embodied Agents

Alan Martin, Gregory M. P. O'Hare, Brian R. Duffy, John F. Bradley & Bianca Schön

Dept. of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland

{alan.martin, gregory.ohare, john.bradley, bianca.schoen}@ucd.ie

brian.duffy@medialabeurope.org

<http://chameleon.ucd.ie/>

Abstract

The notion of embodiment within a virtual environment and an intelligent agent could be intimately linked, in order to form a single Agent Chameleon. This paper presents an architecture for achieving this, while allowing agents freedom to choose an embodiment suitable to the task at hand.

1. Introduction

Over the last number of years, extensive research has been carried out into the area of autonomous agents. The term *agent* can be somewhat nebulous and invoke different meanings depending on the discipline in question. We use the term in a manner synonymous with the Distributed Artificial Intelligence (DAI) community, that is, agents are characterised by the attributes of autonomy, social ability, reactivity and pro-activity [12]. Furthermore, a stronger notion of agenthood also assigns mentalistic attitudes, that is, knowledge, belief, intention and obligation.

In parallel, significant developments in Collaborative Virtual Environments have occurred of late. A number of different systems have sought to incorporate agents within virtual environments. Notably, these include the MAVE system developed by Cobel, Harbison & Cook [3, 4], and the VITAL system developed by Anastassakis et al. [1]. Some important research has also been carried out by Torres et al. [11], and by Huang et al. [7] in integrating BDI based agents within virtual worlds.

Further to this, much research has also been conducted which examines embodiment. *Embodiment* refers to provision of an appropriate representation for an agent or a user within a virtual environment, for the benefit of other agents and users, as well as for the agent itself [2].

This research aims to achieve a fusion of the notion of the embodiment with that of the agent itself, so that the two become inextricably linked. Then the embodiment is

not shallow, as its behaviour is governed by an intelligent process. At the same time the embodiment is free of constraints, and capable of adapting to suit the task at hand. This fusion of the embodied form with the deductive apparatus that governs the behaviour of that form, we term as the *agent chameleon*.

In this paper we propose an Agent Chameleons Architecture for Virtual Environments, a system that empowers agents, facilitating full exploitation of the advantages of virtual environments. The Agent Chameleons project aims to create agents capable of migration between various different platforms, moving seamlessly between between real and virtual environments, mutating their embodied form in order to accomplish their goals effectively.

Section 2 situates this research within the context of the Agent Chameleons project while Section 3 explores some of the motivations for mutation of agent embodiment. Section 4 outlines the architecture of the system that can achieve this and Section 5 expands on this, describing the structure of the agent embodiment within this architecture. Section 6 examines agent deliberation within this system, Section 7 looks at some of the implications of this system, and finally Section 8 looks at an example of an application that utilises the agents freedom of embodiment.

2. Agent chameleons

The Agent Chameleons project [6, 8, 10] endeavours to create the next generation of virtual agents, autonomic entities that can seamlessly migrate, mutate and evolve between and within virtual information spaces. The Agent Chameleon can be seen as a digital *spirit*, capable of occupying and controlling a physical entity such as robot, or an embodied container, like an avatar. This is illustrated in figure 1. The Agent Chameleons may have several applications, for instance as a virtual assistant, following the user between various different spaces and capable of using the advantages of each.

Three key concepts of migration, mutation and evolution

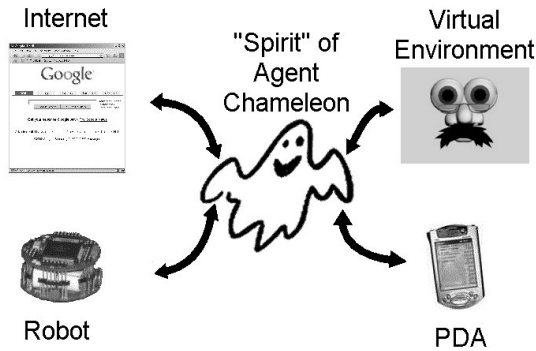


Figure 1. The agent spirit

underpin these agents, allowing them to react to environmental change either by migrating across a network, in order to access a different platform, or mutating or evolving their form in order to improve their capability set. Agent Chameleons are considered as an autonomous, mobile and social entity in the classic multi-agent systems sense. The agent has at any given instance a persona and associated with a given persona are a given set of capabilities. The migration and mutation of agents could thus be invoked in two manners. Firstly by the agent itself, reactive mutation/migration, or secondly by the enveloping environment based upon agent activity, proactive mutation/migration.

The key to this system is the method whereby the agents can migrate between the various information spaces. Agents should be capable of migrating on to a wide variety of devices in order to utilise the features of each. For instance an agent could migrate to a real world robot in order to achieve a physical manifestation and influence physical reality, to a PDA in order to travel with the user, or to a virtual environment in order to improve its abilities for interacting with the user. Ideally the agent would be able to migrate to any given machine, with any given operating system, without any previous support on that machine. However, presently, this is highly improbable, as the agents require somewhere to *live* on any given machine.

We aim to create an Agent Chameleons environment that achieves these requirements. This paper looks at applying this architecture to virtual environments and the possibilities that this presents to the Agent Chameleons.

3. Mutation

Key to maintaining the presence of the agent chameleon within the virtual world is the agents' embodiment, or the nature of the avatar representing the agent. Virtual agents

have traditionally been confined to a single form of avatar, a single embodiment. This research proposes a different approach, our vision is of a system whereby an agent can mutate its form between various different embodiments in order to suit the task at hand.

This provides the agent with a number of advantages. Without the constraints on the embodiment, the agent chameleon is now capable of selecting the embodiment most suitable to its needs. For example, in a space environment the agent could adopt the form of a rocket, allowing it greater freedom of movement. If it requires verbal interaction with the user, the agent could adopt the form of a face, limiting its movement but providing it with the ability to express certain facial expressions.

This freedom provides agents with enhanced flexibility, allowing for an expanded set of uses within the virtual world. Combined with the abilities of evolution and migration between various platforms and information spaces, this helps to realise the vision of Agent Chameleons.

4. Architecture

The selection of an agent's embodiment within a virtual environment is a vital decision for the agent. Different embodiments serve different roles, providing the agent with different abilities in its interaction with the virtual world as well as with the user. Embodiments must serve two purposes, firstly providing the agent with the means to accomplish its task, and secondly preserving the agent's sense of identity within the mind of the user. It is therefore essential that the agent is capable of deliberating upon its choice of embodiment. These requirements are addressed by the Agent Chameleons Architecture for Virtual Environments.

The architecture of this is shown in figure 2. The agents' deliberative mechanism is based upon Agent Factory [5]. Agent Factory provides a cohesive framework for the development and deployment of agent-oriented applications. Specifically, it delivers extensive support for the creation of Belief-Desire-Intention (BDI) agents. The Agent Factory Run-Time Environment delivers support for the deployment of agent-oriented application across a large number of platforms.

Reasoning within agent factory is based upon a series of beliefs. A *belief* is a statement of what the agent believes to be true at the current moment. It is important to realise that these beliefs need not necessarily be true, but the agent believes them to be true and reasons accordingly. A belief "the agent believes that the sky is blue" can be represented by:

```
BELIEF(sky(blue))
```

The behaviour of the agent is defined by commitment rules. They define under which circumstances the agent

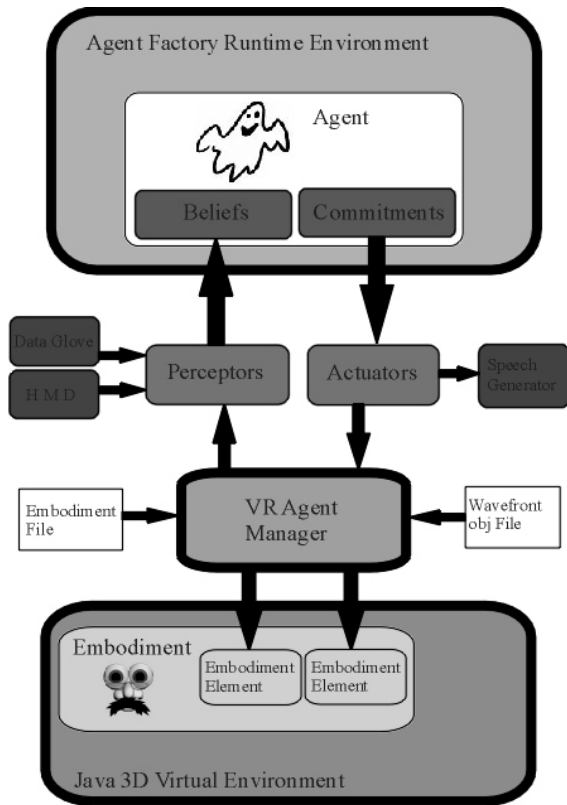


Figure 2. The System Architecture

adopts a particular commitment. *Commitments* represent pledges to assume a course of action as a result of the interpretation of the agent's beliefs with its commitment rules. They represent the outcome of the agent's decision-making process. An agent will commit to raising an umbrella, if it believes that it is raining and it believes that it has an umbrella, if it has the following commitment rule:

```
BELIEF(weather(rain)) & BELIEF(have(umbrella))
=> COMMIT(raise(umbrella))
```

In order to connect these deductive mechanisms with the environment, Agent Factory allows for the creation of actuators and perceptors. *Perceptors* are responsible for the monitoring of the environment within which the agent is operating, and the generation of beliefs based upon it. *Actuators* are responsible for changing the environment in response to the agent's commitments. In this case there are a number of perceptors, perceiving information about the virtual environment and the agent's embodiment within it, as well as other devices connected, such as the position of a data-glove or a head-mounted-display. Actuators are included to influence the virtual world as well as other actions such as generating speech.

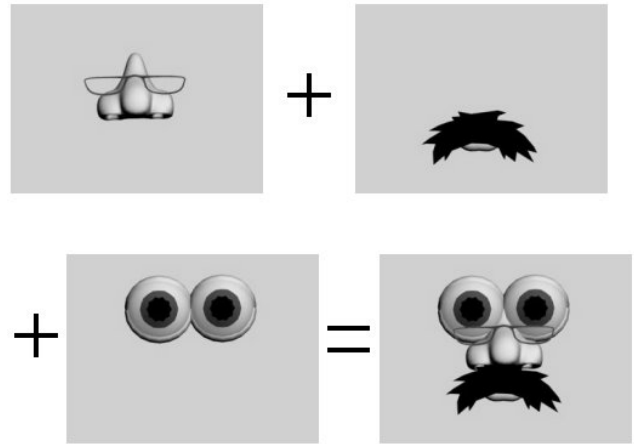


Figure 3. An embodiment consists of a number of embodiment elements

The virtual environment used by Agent Chameleons is based upon Java3D. The VR Agent Manager is responsible for the control of the embodiments within the virtual environment. This receives signals from the actuators and changes the embodiment appropriately. In addition it sends signals concerning the state of the embodiment and the environment to the perceptors.

5. Embodiments

An embodiment can be seen as an aggregation of *embodiment elements*, each of which corresponds to a definite piece of geometry in the avatar. A simple example of this is shown in figure 3, where the embodiment consists of three elements, the eyes, the nose, the moustache and mouth. The main advantage of such a system is that it allows independent animation of the various elements. In this example the eyes could blink or move independently of the motions of the mouth.

The geometry of each of the embodiment elements can be defined using a series of wavefront files. These files must have the same number of objects with the same number of vertices, but can define different positions. Each of these files represents a key frame in the animation of the element, and each is placed at a particular point along a time-line. The time-line is defined using frame numbers, which are interpolated at a rate of thirty frames per second. Based upon this the relationship of the animation can then be defined.

The animations of each element operate based upon a series of animation states. Each element consists of a number of states that define particular times at which the animations overlap. These represent points where the geometry is in a common position, synchronisation points at which

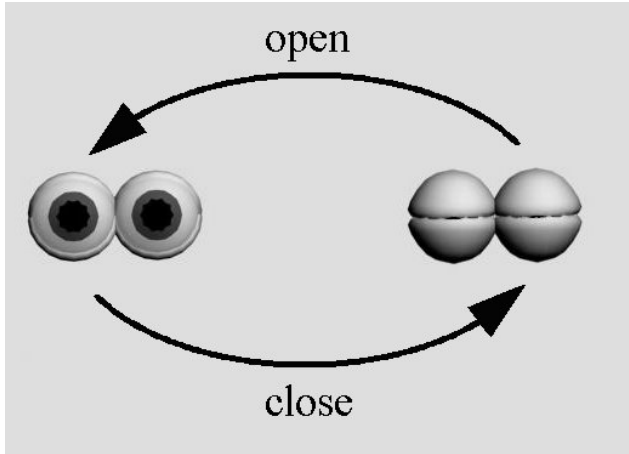


Figure 4. A simple animation state graph

the animations can be stitched together. These are linked to other states by the animations that the element can undergo, forming a directed graph of possible animations. A simple example of this is illustrated in figure 4. In this case there are two animation states, eyes open and eyes closed, with two animation connecting them. The animations can be controlled by the traversal of this graph, executing the required animations as needed.

5.1. Embodiment description file

In order to allow ease of creation and manipulation of the embodiments by an agent designer, each of the elements of an embodiment, their animations and animation states, are defined within an embodiment description file. This is in effect a simple text file, with the embodiment defined using a subset of XML. An example of such a file is shown in figure 5.

A number of things should be noted about the structure of these files. Firstly the animations are defined as a series of steps, each between a number of frames. The frame numbers are a reference to the time-line of the embodiment element. In addition the visibility of elements can also be turned on and off as part of their animation. Secondly, as can be seen from the *nose* element of the example, if an element supports no animation, then animation states do not need to be defined for it. Thirdly, when animation states are defined, an initial state must also be included. This is the state that the avatar will be in when the agent first adopts it as an embodiment, and effectively represents a precondition for the invocation of a particular animation.

An animation ontology such as this allows the embodiment designer ease of control over the form and function of the various embodiments. These files are made available to the VR Agent Manager, which is subsequently capable of

```
<avatarForm name="Face" initialY="+2.1">
  <avatarElement name="eyes"
    initiallyHidden="false">
    <keyFrame frameNumber="0"
      fileName="eyes_0.obj" />
    <keyFrame frameNumber="5"
      fileName="eyes_5.obj" />
    <keyFrame frameNumber="10"
      fileName="eyes_10.obj" />

    <animation name="open">
      <step start="0" stop="20"
        loop="false" />
    </animation>

    <animation name="close">
      <step start="0" stop="20"
        loop="false" />
    </animation>

    <state name="eyesclosed">
      <transition to="eyesopen"
        animation="open">
    </state>

    <state name="eyesopen">
      <transition to="eyesclosed"
        animation="close">
    </state>

    <initialState name="eyesopen" />
  </avatarElement>

  <avatarElement name="nose"
    initiallyHidden="false">
    <keyFrame frameNumber="0"
      fileName="nose.obj" />
  </avatarElement>

  <avatarElement name="mouth"
    initiallyHidden="false">
    <keyFrame frameNumber="0"
      fileName="mouth_0.obj" />
    <keyFrame frameNumber="20"
      fileName="mouth_20.obj" />

    <animation name="say">
      <step start="0" stop="10"
        loop="false" />
      <step start="10" stop="0"
        loop="false" />
    </animation>

    <state name="close">
      <transition to="close"
        animation="say" />
    </state>

    <initialState name="close" />
  </avatarElement>
</avatarForm>
```

Figure 5. A Sample Embodiment Description File

selecting and controlling any of the embodiments supported by the environment.

6. Agent control

When an agent enters the virtual environment, the perceptors generate a number of beliefs, these might typically include:

- Beliefs about the embodiments that can be adopted, e.g. `BELIEF(possibleEmbodiment(Face))`
- Beliefs about the embodiment currently adopted, e.g. `BELIEF(currentEmbodiment(Face))`
- Beliefs about the animations that can currently be executed, e.g. `BELIEF(possibleAnimation(eyes, blink))`
- Beliefs about motion that the agent can make, e.g. `BELIEF(allowedMotion(X))`
- Beliefs about the avatars current location, e.g. `BELIEF(currentLocation(0, 1.4, 0))`
- Beliefs about any devices attached to the system, such as a data glove or head mounted display.

In addition the agent is equipped with several actuators. Agents are capable of committing to a number of different actions, including:

- Mutation of the agents form through the adoption of new embodiments.
- Invocation of animation of one of the elements of the current embodiment state.
- Motion within the world, which may or may not have an animation associated with it, depending upon the embodiment.
- Interaction with the user through any interaction modalities (e.g. voice capabilities).
- Migration to another environment altogether.

Based upon these beliefs and commitments the agent can deliberately control its embodiment. Referring back to the example outlined in figure 5 above, when the agent first adopts the *Face* embodiment it gains the following beliefs:

```
BELIEF(possibleEmbodiment(Rocket))
BELIEF(currentEmbodiment(Face))
BELIEF(possibleAnimation(eyes, close))
BELIEF(possibleAnimation(mouth, say))
```

This means that the agent believes that it has the *Face* embodiment, that it could also adopt the *Rocket* embodiment. It also believes that it can animate the *eyes* with the *close* animation, and the *mouth* with the *say* animation. The agent could now commit to an animation, for instance by committing to the following:

```
COMMIT(triggerAnimation(eyes, close))
```

After this commitment, and the animation has completed, the animation state of the *eyes* element will have changed from *eyesopen* to *eyesclosed*. The agents beliefs regarding the possible animations that it can invoke will therefore be updated, and the agent's beliefs will be:

```
BELIEF(possibleEmbodiment(Rocket))
BELIEF(currentEmbodiment(Face))
BELIEF(possibleAnimation(eyes, open))
BELIEF(possibleAnimation(mouth, say))
```

It is important to note that the animation of the various elements within the embodiment is completely independent. The animation of the eyes has no effect upon the mouth and the mouth no effect upon the eyes. It is therefore possible to reason about the animations separately and to invoke animations on different elements simultaneously.

The agent could also commit to changing its embodiment entirely by issuing the following commitment:

```
COMMIT(adoptEmbodiment(Rocket))
```

7. Identity

The opportunity for an agent to autonomously change its embodiment freely and easily presents a number of identity problems. If the agents can change it's form, how can the notion of the agent be maintained in the mind of the user? This is an essential requirement if the user is to empathise with the agent, and to develop a relationship with it. The sense of identity of the agent must be preserved, across all of the possible embodiments that it can adopt, with common visual cues or themes that are identifiable in all embodiment forms.

Designers of the embodiments must take care to maintain the agents sense of self, ensuring that certain factors remain constant across all embodiments. Common colour schemes or particular face markings could be used in order to do this. Non visual factors such as the tone of voice of the agent could also be used. This sense of identity applies not only to virtual environments, but to other platforms that the agents can occupy. Other platforms such as robots or PDA's could attempt to use the same common features.

8. Case study: scissors-paper-stone

The children's game of Scissors-Paper-Stone (also known as Rock-Paper-Scissors) is an example of how the agent's freedom of embodiment can empower it. It also provides an illustration of how such agents can interact with the user.

The game is played by two players. Each simultaneously selects one of scissors, paper or stone, indicating their choice using hand gestures. A flat palm indicates paper, a clenched fist indicates stone and a two fingers indicates scissors. Scissors beats paper as scissors can cut paper and destroy it, stone beats scissors as stone can smash and damage scissors, and paper beats stone as paper can wrap stone. A draw results in a rematch.

In this case one of the players is an agent, which is initially embodied as a face. In this embodiment, the agent is capable of speech generation and synthesis, through the use of IBM Via Voice and the Java Speech API. The user is equipped with a 5DT Data Glove 5. This features sensors for measuring the position of the fingers. Using this glove the user chooses either scissors, paper or stone. The necessary gesture is recognised and interpreted. Simultaneously, the agent randomly selects one of three possible embodiments, corresponding to scissors, paper and stone, in an attempt to beat the user. In this way the real world user can interact with the agent located within the virtual world. The sense of identity of the agent is maintained by the correlation between scissors, paper and stone within the mind of the user.

An example of the agent's reasoning during a game can be seen in figure 6 (a), (b) and (c). Initially the agent is embodied as a face, as in figure 6 (a), it has beliefs about the possible embodiments of scissors, paper and stone that it may adopt. When the user initialises a game, the agent commits to adopting one of the embodiments, in this case scissors. Once the new embodiment has been adopted, as shown in figure 6 (b), the agent gains beliefs about the possible animations that it can execute. It triggers the animation to open the scissors, this results in a change in the animation state of the scissors, and hence a change in the possible animations, as in figure 6 (c).

After illustrating its choice with the selection of its embodiment, the agent readopts the face embodiment in order to tell the user who won, and to allow the game to be played again.

9. Conclusion

Virtual Agents are much flexible when they are unconstrained by their embodiments. Our vision is of agents free to mutate their embodiments in order to take full advantage

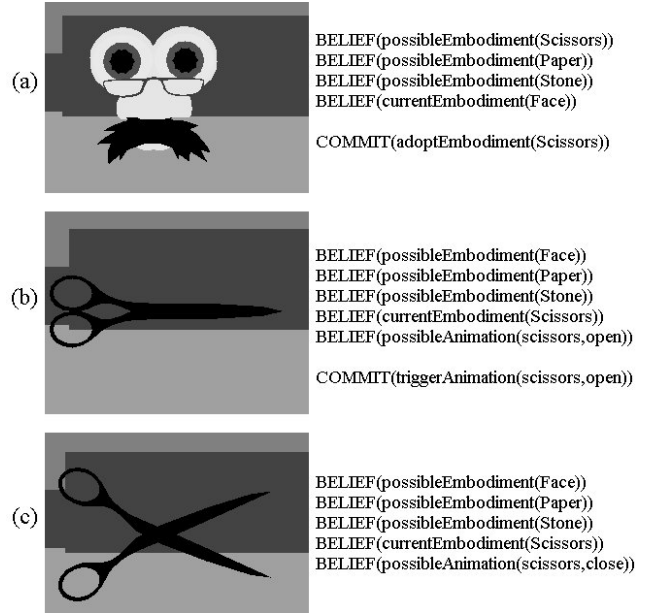


Figure 6. Agent Embodiment and Mental State During Game

of them. However, throughout these embodiments, it is important that the agents maintain a sense of identity.

The Agent Chameleons Architecture for Virtual Environments can help to achieve this. The architecture allows agent full control over their embodiments, allowing them to mutate their forms at will. These changes are fully controlled by the agents deliberative mechanism, and hence the agents and embodiments become fused to form a single agent chameleon. The architecture also allows the agent designer to easily and effectively create and modify the possible embodiments of the agent.

Acknowledgements The work undertaken as part of the Agent Chameleons project (<http://chameleon.ucd.ie>) is a collaborative project between the Department of Computer Science, University College Dublin (UCD) and Media Lab Europe (MLE), Dublin. We gratefully acknowledge the financial support of the Higher Education Authority (HEA) Ireland and the Irish Research Council for Science, Engineering and Technology: funded by the National Development Plan.

References

- [1] G. Anastassakis, T. Ritchings, and T. Panayiotopoulos. Multi-agent systems as intelligent virtual environments. In *Advances in Artificial Intelligence, Joint German/Austrian*

Conference on AI - KI 2001, volume 2174 of *Lecture Notes in Computer Science*. Springer, September 2001.

- [2] S. Benford, J. Bowers, L. E. Fahlén, C. Greenhalgh, and D. Snowden. User embodiment in collaborative virtual environments. In *Proceedings of CHI 1995*, 1995.
- [3] J. Cobel and D. J. Cook. Virtual environments: An agent-based approach. In *Proceedings of The AAAI Spring Symposium on Agents with Adjustable Autonomy*, 1999.
- [4] J. Cobel and K. Harbison. MAVE: A multi-agent architecture for virtual environments. In *Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 577–583.
- [5] R. Collier, G. M. P. O’Hare, T. Lowen, and C. Rooney. Beyond prototyping in the factory of agents. In *Proceedings of The Third Central and Eastern European Conference on Multi-Agent Systems - CEEMAS’03*, Prague, Czech Republic, 2003.
- [6] B. R. Duffy, G. M. P. O’Hare, A. N. Martin, J. F. Bradley, and B. Schön. Agent chameleons: Agent minds and bodies. In *Proceedings of Computer Animation and Social Agents - CASA 2003*, May 2003.
- [7] Z. Huang, A. Eliëns, and C. Vissser. Programmability of intelligent agent avatars. In *Proceedings of Agent’01 Workshop on Embodied Agents*, Montreal, Canada, June 2001.
- [8] G. M. P. O’Hare and B. R. Duffy. Agent chameleons: Migration and mutation within and between real and virtual spaces. In *Proceedings of The Society for the Study of Artificial Intelligence and the Simulation of Behaviour (SSAISB) - AISB’02 Conference*, April 2002.
- [9] G. M. P. O’Hare, B. R. Duffy, J. F. Bradley, and A. N. Martin. Agent chameleons: Moving minds from robots to digital information spaces. In *Proceedings of Autonomous Minirobots for Research and Edutainment - AMiRE 2003*, February 2003.
- [10] G. M. P. O’Hare, B. R. Duffy, B. Schön, A. Martin, and J. F. Bradley. Agent chameleons: Virtual agents real intelligence. In *Proceedings of The Fourth International Working Conference on Intelligent Virtual Agents - IVA 2003*, volume 2792 of *Lecture Notes in Computer Science*, pages 218–225, Irsee, Germany, September 2003.
- [11] J. A. Torres, L. P. Nedel, and R. H. Bordini. Using the bdi architecture to produce autonomous characters in virtual worlds. In *Proceedings of The Fourth International Working Conference on Intelligent Virtual Agents - IVA 2003*, volume 2792 of *Lecture Notes in Computer Science*, pages 197–201, Irsee, Germany, September 2003.
- [12] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2), 1994.